

**Amendments to the Claims:**

1 (currently amended): A computer program product embodied on a computer-readable storage medium and comprising code stored on the computer-readable storage medium, the code such that, when executed by a processor, causes a computing device to perform the following:

receiving information from a client computing device at a server component on a server computing device, the server computing device configured for dynamic allocation of buffer memory on the server, the buffer memory on the server to be allocated to clients for file system transactions, wherein the information indicates the client needs additional resources to perform a transaction and the information received from the client includes a number of transactions that are currently pending on the client but have not been processed that exceed a maximum number of transactions available limit that was previously negotiated;

determining by the server component if allocating to the client the additional buffer memory on the server puts the server component in a resource constrained situation, wherein the server component is determined to be in a resource constrained situation by comparing a total number of transactions currently in use for all connections to the server and a total number of transactions that are currently pending on the clients ~~requests~~ for all connections to the server with a maximum number of transactions available on the server;

in response to determining that allocating to the client the additional resources puts the server component in the resource constrained situation:

determining resources currently allocated to a plurality of existing clients;  
wherein the server component stores server-side information related to each client connection with each of the clients, the server storing server-side information for each connection, the server-side information including:

a current number of outstanding transaction requests from the client;  
the maximum number of transactions available limit for the client;  
the number of transactions that are currently pending on the client that exceed the maximum number of transactions available limit for the client,  
wherein the maximum number of transactions available limit for the client is

initially determined when each of the clients connects to the server at which point a negotiation is performed between the client and the server to establish the maximum number of transactions; wherein the maximum number of transactions specifies a number of transaction requests to be accepted by the server from the client;

issuing rebalancing messages ~~by a Light Weight Input/Output (LWIO) protocol~~ configured for distributed file systems to any affected clients to either reduce or increase their maximum transaction available limit, wherein the rebalancing messages to the affected clients comprise deltas, each delta specifying a change in the maximum number of transactions available to the corresponding affected client; wherein the delta is a positive change or a negative change.

2 (original): The computer-readable medium of claim 1, wherein the server component executes on a server in a network environment.

3 (original): The computer-readable medium of claim 1, wherein the server component is further configured to allocate the client the additional resources needed if the server determines that such allocation does not create the resource constrained situation.

4 (original): The computer-readable medium of claim 1, wherein the clients and the server component communicate using a light weight input/output protocol.

5 (currently amended): The computer-readable medium of claim 1, wherein each delta is based on a determination of a credit limit for the client scaled by a completion factor, wherein the completion factor is used to control how many credits are reclaimed even if not used.

6 (currently amended): The computer-readable medium of claim 5, wherein the delta of each rebalancing message is ~~restricted to~~ a one credit delta.

7 (previously presented): The computer-readable medium of claim 6, wherein the delta of each rebalancing message comprises a plurality of credits.

8 (currently amended): The computer-readable medium of claim 1, wherein the rebalance of the resources is performed based on ~~a an equitable~~ equitable distribution of the resources among the plurality of clients, wherein the ~~equitable~~ equitable distribution allocates server buffer memory from an existing client to a new client if the existing client is not using the server buffer memory.

9 (currently amended): The computer-readable medium of claim 8, wherein the ~~equitable~~ equitable distribution of the resources is based on a number of clients connected to the server component.

10 (previously presented): The computer-readable medium of claim 9, wherein at least one client connection is assigned a higher priority than connections of other clients.

11 (currently amended): The computer-readable medium of claim 8, wherein the ~~equitable~~ equitable distribution of the resources is based on a number of open files associated with each client connected to the server component.

12 (previously presented): The computer-readable medium of claim 11, wherein at least one open file is assigned a higher priority than other open files.

13 (currently amended): A computer program product embodied on a computer-readable storage medium and comprising code that, when executed, causes a computing device to perform the following:

a plurality of data stores, each data store being associated with a different client connection to a server computing device, wherein the server computing device is configured for dynamic allocation of buffer memory on the server, each data store including:

a credits used field that identifies a number of resource credits currently in use by a client computing device corresponding to the data store;

a credit limit field that identifies a number or resources available to the client corresponding to the data store;

a pending count field that identifies a number of transactions that are pending on the client due to an unavailability of sufficient resources to handle the transactions; and

an open files field that identifies a number of files that are currently in use by the client; receiving a transaction request message on the server computing device from the client; wherein the transaction request message received from the client includes the number of transactions that are pending on the client due to an unavailability of sufficient resources to handle the transactions that was previously negotiated; wherein the transactions that are pending on the client have not been sent to the server, wherein the number of resources available to the client that are stored in the credit limit field is a maximum number of transactions available to the client that is initially determined when the client connects to the server at which point a negotiation is performed between the client and the server to establish the maximum number of transactions; and wherein the server rebalances resources when the transaction request places the server in a resource constrained situation as determined in part by a number of transactions that are pending on the clients; and

sending rebalancing messages by a Light Weight Input/Output (LWIO) protocol used for distributed file systems to any affected clients to either reduce or increase their maximum transaction available limit.

14 (original): The computer-readable medium of claim 13, wherein the data store further comprises a flag field that identifies whether the corresponding client has acknowledged a resource-related message.

15 (original): The computer-readable medium of claim 13, wherein a value of the pending count field is provided by the client in connection with a transaction request message.

16 (original): The computer-readable medium of claim 15, wherein a value of the credit limit field is modified based on the value of the pending count field.

17 (currently amended): The computer-readable medium of claim 13, wherein values for the credit limit fields of the plurality of data stores is rebalanced based on a an equitable distribution of available resources.

18 (currently amended): A computer program product embodied on a computer-readable storage medium and comprising code that, when executed, causes a computing device to perform the following:

a server component, the server component configured for dynamic allocation of buffer memory on a server computing device, the server component configured to:

receive information from a client that indicates the client needs additional buffer memory on the server to perform a transaction; wherein the information received from the client includes a number of transactions that are that are pending on the client but have not been sent to the server due to an unavailability of sufficient resources to handle; wherein the number of transactions was previously negotiated; and to rebalance resources currently allocated to the client; wherein the server issues messages to any affected clients when the buffer memory on the server is rebalanced by the server; wherein the messages indicate to either reduce or increase each of the affected clients number of transactions, the messages comprising deltas specifying changes in the maximum number of transactions;

wherein the client maintains information about the state of its allocated resources and pending transactions within a data structure, comprising:

a credits used field that identifies a number of resource credits currently in use by a client corresponding to the data structure;

a credit limit field that identifies a number of resources available to the client; wherein the number of resources available to the client is initially determined when the client connects to the server at which point a negotiation is performed between the client and the server to establish the number of resources;

a pending count field that identifies the number of transactions that are pending due to an unavailability of sufficient resources to handle the transactions; and

a pending queue field that includes transaction messages corresponding to the transactions that are pending.

19 (currently amended): A computer-implemented method embodied on a computer-readable storage medium, that when executed, causes a server computing device configured for dynamic allocation of buffer memory to perform the following:

computing a total number of client connections, each client connection being associated with a client connected to a server, each client having a credit limit stored on the client and the server that identifies a number of resources that are allocated to the client; wherein the number of resources that are available to the client is initially determined when the client connects to the server at which point a negotiation is performed between the client and the server to the number of resources; wherein the client maintains information about the state of its allocated resources including a current number of outstanding credits used and a maximum number of credits available;

computing a total number of pending requests on each client device that have not been issued to the server that identifies a number of transaction requests that are not being handled due to a limitation on resources;

computing a total number of credits in use; and

if the total number of pending requests and the total number of credits in use combined exceeds a total number of available resources, calculating on the server a new credit limit for each of the clients connected to the server;

reallocating the total available resources in accordance with the new credit limits; and

issuing messages by a Light Weight Input/Output (LWIO) protocol configured for distributed file systems to affected clients indicating to either reduce or increase their negotiated number of resources.

20 (original): The computer-implemented method of claim 19, wherein the reallocation is based on each client connection receiving a pro rata share of the total available resources.

21 (original): The computer-implemented method of claim 20, wherein the pro rata share of the total available resources is based on the total available resources divided among the total number of client connections.

22 (original): The computer-implemented method of claim 21, wherein the total available resources are divided evenly among the total number of client connections.

23 (original): The computer-implemented method of claim 21, wherein at least one of the client connections is weighted more heavily than another of the client connections.

24 (original): The computer-implemented method of claim 20, wherein the pro rata share for a particular client is based on a proportion of a total number of open files to a number of open files for the particular client.